

LANGLY
GRANT
1N-16-R
183402
49B

FINAL REPORT

Continued Development of the Space Transportation
Model Simulation on a Microcomputer

Neal M. Bengtson

January, 1989

NASA-Langley Research Center grant NAG-1-792

(NASA-CR-184616) MICROCOMPUTER VERSUS
MAINFRAME SIMULATIONS: A CASE STUDY Final
Report (North Carolina State Univ.) 49 p

CSCL 22B

N89-15148

Unclas
G3/16 0183402

Continued Development of the Space Transportation
Model Simulation on A Microcomputer

The research was conducted to two parts. Part one consisted of a study of the feasibility of running the Space Transportation Model simulation on an office IBM-AT. The second part was to design simulation runs so as to study the effects of certain performance factors on the execution of the simulation model.

The results of this research are given in the two reports which follow: "Microcomputer vs. Mainframe Simulation: A Case Study" and "Fractional Factorial Designs of Simulation Runs for the Space Transportation System Operations Model."

In the first part, a DOS batch job was written in order to simplify the execution of the simulation model on an office microcomputer. A comparison study was then performed of running the model on NASA-Langley's mainframe computer vs. running on the IBM-AT microcomputer. This was done in order to find the advantages and disadvantages of running the model on each machine with the objective of determining if running of the office PC was practical. The study concluded that it was.

The large number of performance parameters in the Space Transportation model precluded running a full factorial design needed to determine the most significant design factors. The second report gives several suggested fractional factorial designs which require far fewer simulation runs in order to determine which factors have significant influence on results.

MICROCOMPUTER VS. MAINFRAME SIMULATIONS: A CASE STUDY

**Neal M. Bengtson
Department of Computer Science
Department of Industrial Engineering
North Carolina State University
Raleigh, North Carolina**

May 1988

**This research was supported by NASA-Langley Research Center under
Research Grant NAG-1-792.**

MICROCOMPUTER VS. MAINFRAME SIMULATIONS: A CASE STUDY

by

Neal M. Bengtson
Department of Computer Science
Department of Industrial Engineering
North Carolina State University
Raleigh, North Carolina

Abstract

A comparison was made of using a simulation language to run models on a mainframe computer versus a microcomputer with a hard disk. The study was performed at NASA Langley using both the SLAM II mainframe and PC versions. The procedure for executing SLAM II on a PC is given. A batch job was created to simplify this procedure. This batch job allows PC's with a hard disk to execute simulations with one command. NASA's Space Transportation System Operations Model and the examples in the SLAM II text were used as a basis for the comparison. The PC simulations completed in predictable times which were almost always faster than the more unpredictable mainframe times. The batch job created insured that the PC runs were as easy to perform as mainframe runs.

Introduction

The use of simulation models to study the operational characteristics of future space transportation systems has become a standard practice in the conceptual studies at NASA Langley. However, the use of the proprietary language SLAM II, available

only for mainframes when first leased in 1980, has become cause for concern due to impediments in operation and to rising fees and user charges. The recent availability of a full implementation of the language for a microcomputer raised the possibility of running programs on a personal computer and thus avoiding some of the problems currently being encountered with the mainframe usage at Langley. A number of questions are raised with this possibility that needed resolution to have a full understanding of each system's benefits and constraints. What is the effect of program size and run time on the choice of mainframe or PC version? Where do NASA's programs fall in this spectrum and if PC use is possible how will the PC capacity be exceeded in the near future? There are many additional questions. The results of this study represent an attempt to address these uncertainties.

Space Transportation Model

The Space Transportation System Operations (STS Ops) Model [1] was used as a representative program to evaluate and compare the use of the mainframe and microcomputer to perform simulation studies. This is a program designed to assess the operational requirements of a space transportation system operating in both the earth and space environment (see Figure 1). As configured for the study, the program provides for the delivery of payloads from earth to the space station, servicing at the station, and delivery beyond the station using space based transportation elements. The program is divided into three major modules

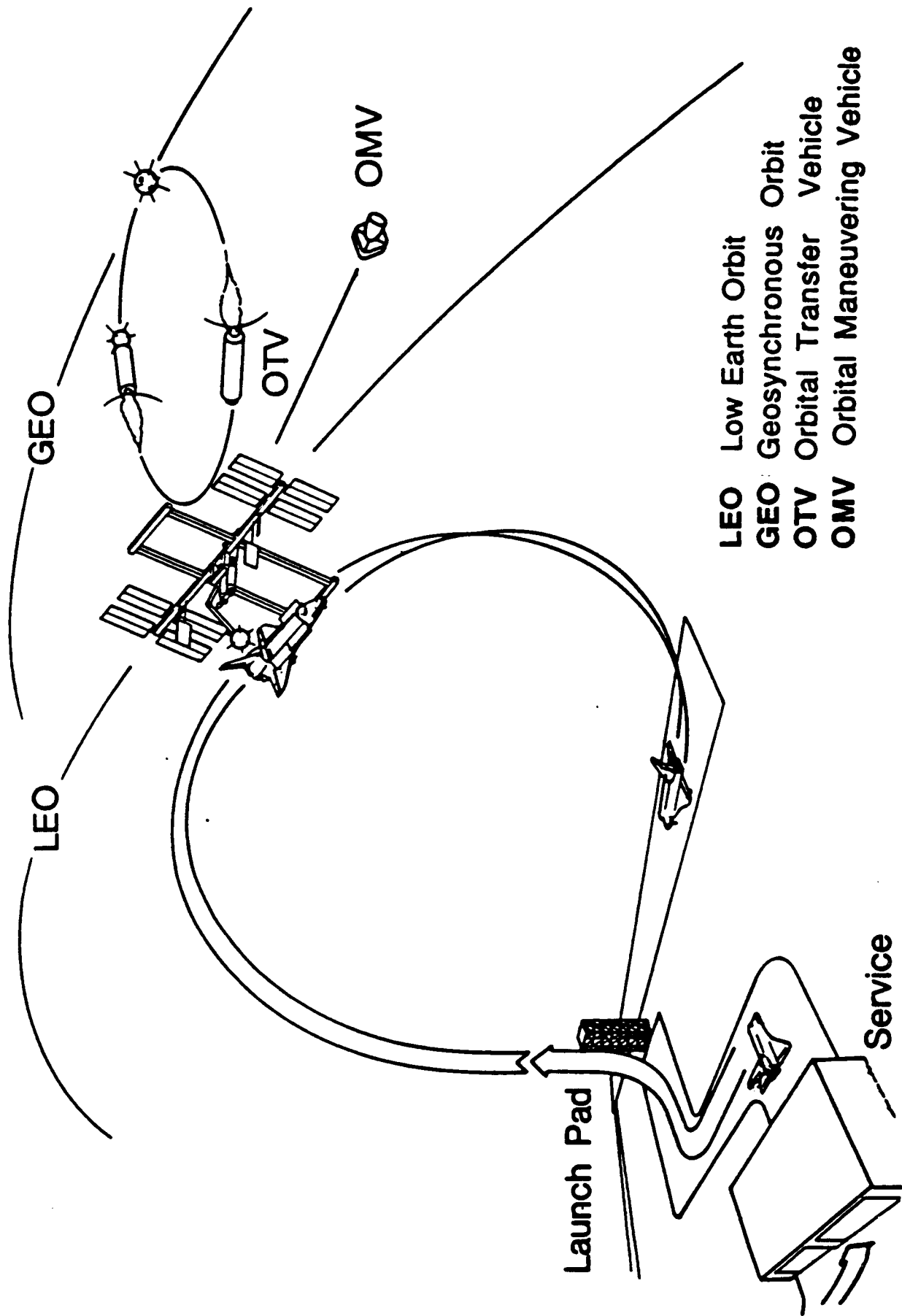


Figure 1.- Transportation System Operating Scenario.

dealing with the ground-based operations, space station operation, and on-orbit operations (see Figure 2). The model is designed to allow the users to enlarge or modify the model to study their area of interest and in this manner maintains a large amount of flexibility.

The simulation model is comprised of three parts: a data input file to drive the model's delivery requirements, a network model which describes the overall flow of vehicles and payloads through the model scenario, and a FORTRAN coded event file to handle model logic too complex to be conveniently handled in the network code. As set up for the standard scenario used in making the comparisons, the data file, 29,000 bytes in size, contained 147 payload entries, the network model was about 11,000 bytes long, and the event file contained about 15,000 bytes of code. This represented a typical scenario to cover a 5 year period of space operations and was run in a deterministic manner for sensitivity studies.

The PC version of SLAM II

SLAM II [2] was the simulation language used for modelling and to make this comparison study. It is a combined network, discrete event and continuous simulation language which is FORTRAN based. Network simulation models are defined with SLAM II network statements that represent different possible operations of a system, for example queues and service activities. Specific execution and control details, such as run length, job name, and number of runs, are given by SLAM II

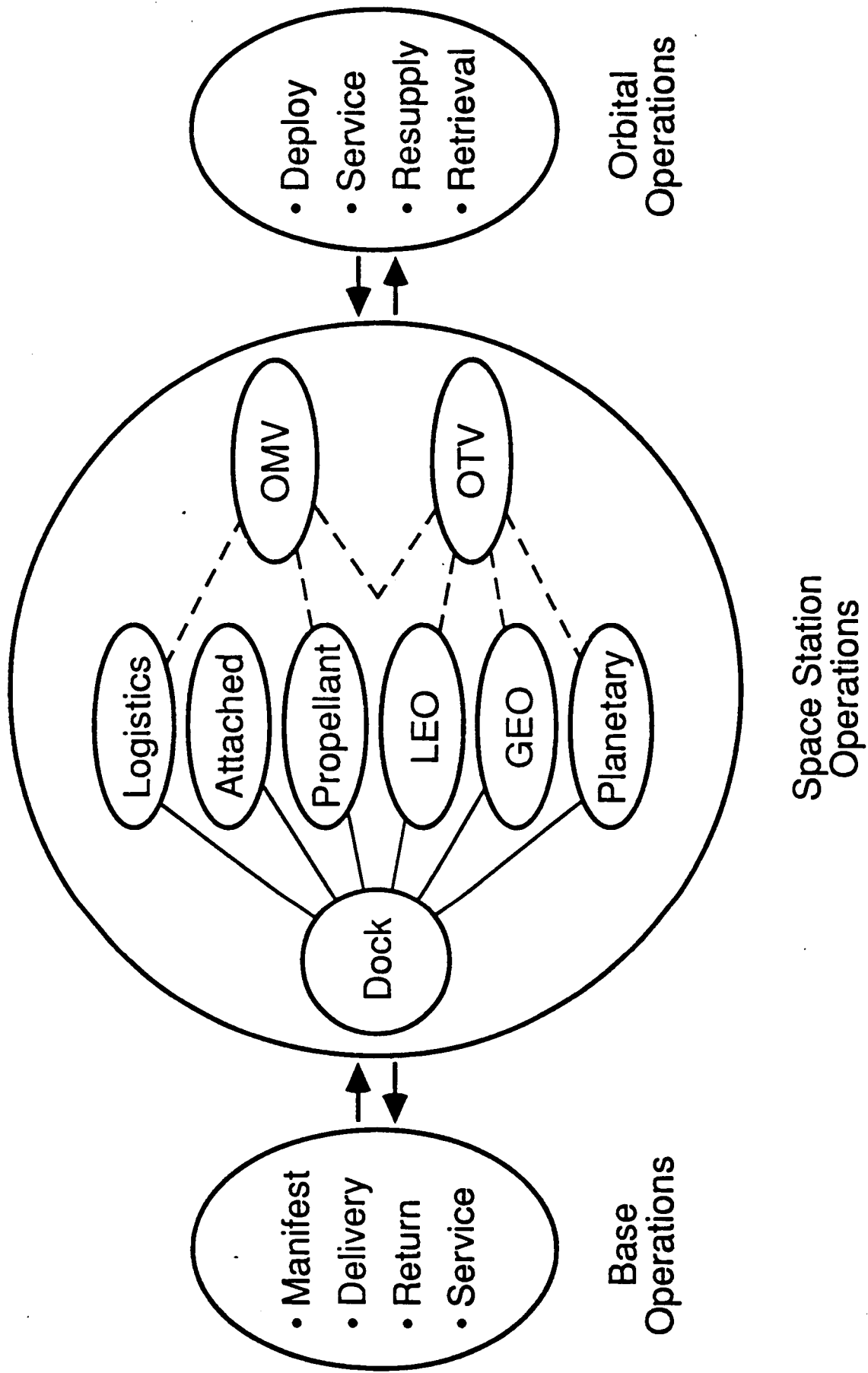


Figure 2.- Schematic of the Space Transportation System Operations Model.

control statements. Any network simulation may be supplemented with FORTRAN written discrete event subroutines to allow for situations in a model which may not be simulated easily by a network formulation. Using the FORTRAN capacity of SLAM II requires a FORTRAN compiler. This compiler is not supplied with SLAM II.

The PC SLAM II version 3.0 used in this study came from P&A (Pritsker & Associates, Inc.) as a four disk pack of 5.25 inch double sided, double density diskettes. For a simulation with a network model and user written FORTRAN like the space station simulation of this study three of the four disks were needed. The first disk contained two processors, INPUT and EXECUTION, the second disk had an OUTPUT processor. INPUT reads a file that contains the SLAM II control statements and network description, checks these for syntax, and makes a translated model. The EXECUTION processor runs a simulation using the translated model for a network only simulation. The OUTPUT processor puts the output from the EXECUTION processor in a readable form. For a simulation with user written FORTRAN subroutines a third disk was used. That disk contained a SLAM II object library which is used with Microsoft FORTRAN [3]. The fourth disk contained another SLAM II object library for use with another version of Microsoft FORTRAN, therefore, only one of the last two disks was needed for any particular simulation.

A new PC version (version 4.0, available October 1987) of SLAM II has been released since the work described here was

performed. This version can only be used with a new version of Microsoft FORTRAN, also version 4.0. The new SLAM II version uses a batch mode similiar to the batch job created for this study. The SLAM II files have been changed to be more flexible and to make executing a simulation easier. In addition, the new FORTRAN version uses a different, and easier, procedure for compilation and linking. Both these languages use more diskettes than before and they both recommend using a hard disk machine. The new version of SLAM II is a validation of the batch job procedure developed during the simulation study of the Space Transportation Model. Before going over the comparison of running the STS Ops Model on the mainframe with the microcomputer, the development of batch jobs created to execute the model on an IBM AT will be given. Without a simplified procedure, it was felt that running SLAM II jobs which required FORTRAN subroutines was too cumbersome a prospect to make microcomputer operation an attractive alternative. Those still using the earlier PC versions of SLAM II will find it a worthwhile illustration of how they can use their existing language more efficiently. The users of the current version of SLAM II will find even more advantage to performing simulations on a microcomputer then found in this study.

Running a PC Simulation

Details for running the PC version of SLAM II are supplied by P&A in a user's manual [4]. Below will be given a brief description of the procedure for executing a combined

network-user written FORTRAN simulation such as the STS Ops. Model. This will give an idea of the rather elaborate process necessary to run a complete SLAM II simulation with FORTRAN routines on a personal computer. The objective is to motivate the creation of a batch job which takes advantage of the hard disk and performs many of the steps needed to execute a SLAM II program. By going through this program execution process, the listing of the batch job created (see Appendix B) should be easier to follow and understand.

A combined network and user written subroutines simulation, such as the Space Transportation Model, is more complicated to run than a network only simulation. In addition to the three SLAM II disks mentioned above, four Microsoft FORTRAN disks were needed. These were the ones used for the first two FORTRAN passes, as well as ones that contained the FORTRAN library, math library, and the linking routine. A disk for the SLAM II control and network statements, the FORTRAN routines, space for output, and, if needed, a data file was also needed. These simulation model files may take more than one disk. This means eight or more disks were needed to produce a simulation run. By copying files onto new disks, some consolidation may be made and two or three disks eliminated from the process. Even so, there is still a great deal of disk swapping and typing needed when running the simulation. This increases the time it takes to get a completed run.

Before attempting to run a SLAM II program, a file was

created which contained the SLAM II control and network description statements. A second file was created which had the user written FORTRAN subroutines. Once these files were created, with the INPUT and EXECUTION disk in drive A of the PC, "INPUT" was typed to begin. When prompted, the statement file name was typed. Upon successful translation a file name was entered where the translated file was to be stored.

The SLAM II disk containing the object library for SLAM II which was to be joined with a FORTRAN library and a math library for linking with the user written FORTRAN routines was used next. The first Microsoft FORTRAN disk performed a translation of the FORTRAN routines. The second FORTRAN disk created a relocatable object file for linking with the libraries mentioned above which were on a third Microsoft disk. A fourth Microsoft disk contained the linking program that puts all this FORTRAN together on an executable file. Before execution could be attempted a set of meta commands from Microsoft had to be added to the beginning of any file containing FORTRAN routines. These commands were provided by P&A in a file, but needed to be altered slightly depending on the requirements of the FORTRAN routines. A SLAM II MAIN program was also provided for linking so that unless changes are need in MAIN one need not be included in the user written routines. The next step was to use the first two Microsoft disks to create object files of the user written FORTRAN and MAIN routines. These object files were created separately. A linking program was run next to link the two object files with the SLAM

II, FORTRAN, and math libraries. The executable file created by this process was then run. In response to a prompt, the name of the network translated file was entered. The simulation was then performed. After completion, another prompt asked for the name of an output file for storage of the SLAM II results.

At this point it was necessary to replace the INPUT and EXECUTION disk with the OUTPUT processor disk. After "OUTPUT" is typed the output file name was requested. A menu then appeared from which the user selected all or part of the standard SLAM II summary report desired. A second menu appeared to indicate where this output was to be sent. After reviewing the simulation results, the user must return to the first menu from where another segment of output may be requested, or exit out of the OUTPUT processor.

The description above should make it obvious that without a hard disk the job of running a SLAM II simulation with FORTRAN subroutines is too impractical if an alternative is available. The question then is: can the PC version of SLAM II be made more convenient to execute by taking advantage of a hard disk?

Using the Hard Disk

The minimum space necessary to run a SLAM II version 3.0 program with FORTRAN routines is shown in Figure 3. There are eleven files needed (not including files SLAM II.BAT and KEYFAKE.COM). These take a total of 1,161,096 bytes of disk space. Together with the space needed by the simulation program itself and the space needed to store the large

<u>FUNCTION</u>	<u>FILE</u> (bytes used)	<u>FUNCTION</u>
SLAM	INPUT.EXE (146506)	Translates the SLAM control and network statements into a form that will be executed and checks for SLAM syntax.
	EXECUTIO.EXE (168902)	Runs the simulation using as input the translated model from INPUT.EXE and the FORTRAN file created by the linking program.
	OUTPUT.EXE (108926)	Puts the output of EXECUTIO.EXE into a form that can be called from a menu.
	SLAM32.LIB (352256)	The SLAM object library that is used in linking with user written FORTRAN routines. (This is for use with Microsoft FORTRAN version 3.2. Other SLAM object libraries are available for use with versions 3.1 and 3.3.)
	MAIN.OBJ (638)	The SLAM main program object file created from MAIN.FOR supplied with the SLAM disks. This is created by FORTRAN and after creation it is no longer necessary to save MAIN.FOR.
FORTRAN	FOR1.EXE (119890)	Runs the first pass of the Microsoft FORTRAN compiler on the user written FORTRAN routines.
	PAS2.EXE (97152)	Runs the second pass of the FORTRAN.

Figure 3: The files in directory SLAM.

	FORTRAN.LIB (85504) 8087.LIB (18432)	Libraries linked with SLAM32.LIB to the MAIN program and the user written FORTRAN routines.
	LINK.EXE (39680)	The program which performs the linking operation of the FORTRAN object files.
SYSTEM	COMMAND.COM (23210)	Contains DOS system library.
BATCH	SLAMII.BAT (15451)	The batch file used to link and run all the program files automatically.
	KEYFAKE.COM (640)	A program called by SLAMII.BAT that inputs responses to the SLAM processors automatically.
TOTAL	(1177187)	

Figure 3: The file in directory SLAM (continued).

intermediate files four double sized, double density disks is a theoretical minimum number of disks needed. However, because of file partitioning and other practical considerations five disks is a more realistic number. In fact, a five disk set has been created for running SLAM II Version 3.0 simulations, network only or combined.

Since the total file space required for the execution files is less than 1.2 million bytes a 1.2 million byte, high density disk can store all these files. At least one other disk is still necessary for the program and intermediate files. A better approach is to use a hard disk. Once all the run files are on a hard disk with plenty of extra working space, a batch job can be created which will read the input files, create all the intermediate files, perform the FORTRAN compilation, link the object files, execute the simulation, and run the OUTPUT processor stopping at the output menu. Such a batch job was created. This is the job that was used to make the comparisons of running SLAM II simulations on a microcomputer, in this case an IBM AT, vs. a mainframe.

The objective of creating a batch job was to construct a program which would perform automatically all the necessary steps of running a SLAM II program on the IBM AT (See Appendix A). The job created requires one input parameter. It will automatically run either network only or combined simulations. The FORTRAN routines may read data from another file. If more than one run is performed or if intermediate summary reports are

written these will be automatically stored in separate output files and the first one will be used by the OUTPUT processor. The batch job created to do this is one which has been found to work well. There are, of course, other ways to construct this job.

Two directories are used in drive C, the hard disk, of an IBM AT (or XT or PS/2). The first directory contains the files listed in Figure 3. These are the SLAM II, FORTRAN, DOS, and batch files needed to run SLAM II simulations. This directory is called "SLAM". The other directory, called "SLAMJOBS", contains the user's simulation program files. One simulation may use up to three files. One file is the SLAM II network and control statements, another the user written FORTRAN, and the third a data file for data which is read in by the FORTRAN routines. This third file would be used if the FORTRAN data is stored as if it were on a separate unit. In this case, the unit must be "opened" before being addressed by the FORTRAN. Data may also be read from the SLAM II file in the normal way. These three files must all have the same filename. They are distinguished by their extenders. These must be .SLM for the SLAM II network and control file, .FOR for the FORTRAN file, and .DAT for the data file.

The Batch Job

The batch job that runs SLAM II is called SLAMII.BAT. It is set up to run from either directory. However, the intention is to have runs made from directory SLAMJOBS. The output files will

be placed in the directory from which SLAMII.BAT is called. Upon completion, the user will be left in directory SLAMJOBS. In order not to change the contents of directory SLAM the simulation should be executed from SLAMJOBS. A more detailed description of SLAMII.BAT and a listing is given in Appendix B.

As an example of how to run a simulation, assume a simulation program is written called EXAMPLE. This program has user written FORTRAN subroutines and data stored in separate files. There would then be three files: EXAMPLE.SLM, EXAMPLE.FOR, and EXAMPLE.DAT. The user takes the following steps.

1. Puts the files EXAMPLE.SLM, EXAMPLE.FOR, and EXAMPLE.DAT into directory SLAMJOBS.
2. Runs the batch job by typing "SLAMII EXAMPLE". The batch job will now run until the job is complete and the output menu is displayed.
3. Uses the SLAM II output menu to review and store the results.

Comparison of PC and Mainframe

Both the mainframe and PC procedures created for SLAM II programs use three files to run the STS Ops Model. When using the mainframe the FORTRAN routines file is loaded first. The SLAM procedure is called which compiles the FORTRAN and loads SLAM. Next, the file containing data is loaded. Finally, the simulation is run. Output is stored in a file which may then be printed. This is similar to the procedure for running on a microcomputer. There the three files are stored in a directory and used automatically by the SLAM batch job. There may be more than one output file with the microcomputer procedure and these

would have to be examined separately. The separate output files could be loaded into one file and that file printed in order to avoid multiple printouts. (See Appendix C for a batch job which does this.)

Run Results

In order to obtain times to perform steps in the run process on the IBM AT, the batch job which runs the simulation processors was modified to record times in a separate file at selected points during the run. As would be expected the run times on the AT were almost constant for multiple executions of the STS Ops Model. The same could not be said for the model run on the mainframe. A dayfile was saved and examined for times of runs made at various times on the day.

The mainframe used was the CDC Cyber 175 at the NASA Langley facility. The ten best turn around times recorded for the mainframe ranged from 5.7 to 19.5 minutes with an average of 14.2 minutes. The times were highly dependent on the system load at the time of execution. Runs have taken as much as 1 hour, but generally, if the run takes more than 30 minutes the user logs off and tries again latter. The turn around time for the STS Ops Model on the IBM AT is a relatively constant 6.3 minutes.

In order to get further comparisons, all the examples from the SLAM II text [2] were run on both the IBM AT and mainframe computers. These problems included nine network only simulations and nine problems with FORTRAN routines. On the IBM AT the network simulations took .5 to 1.7 minutes and the

combined problems took 3 to 4.2 minutes. The mainframe turn around times were generally 2 to 3 times as long although some runs were much longer. The mainframe never took less time than the IBM AT.

Comparison Observations

Among the advantages usually associated with use of a mainframe computer as compared to a PC are faster execution speed, greater storage capacity, greater array dimension capacity, editing convenience and easier program set up for running.

Using the mainframe to run SLAM II programs at NASA Langley has produced problems that prompted this comparison study. The mainframes are very heavily utilized. Many of the users have programs that require large capacity, fast machines. The mainframe computer is very efficient in handling a large number of users but, from an individual user's point of view, inconveniences can result. The biggest problem encountered by an individual user with the mainframe computer is its unpredictability. Often, just in attempting to log on the mainframe access can be denied because of machine congestion. This forces the user to try again later or to wait. The length of wait is not predictable. Once the user is given access to the mainframe, further delays are always encountered while waiting for responses to the commands necessary to run a program. These delays are frustrating because the terminal must be monitored while waiting for responses. Since the response

times are unknown beforehand, the user may be forced to stay at the terminal for a lengthy period. If the user gives up and tries again later, the time already spent is largely wasted. Another problem which has occurred is that when the mainframe is not operating there is no alternative but to wait for it to become active again. Repair times for a mainframe are also not predictable.

In order to continue using SLAM II on the mainframe a yearly leasing fee must be paid. This fee has been set at 25% of the current initial leasing cost. When this study was done the yearly leasing cost was \$1575. This fee had risen from \$1000 in 1982. The yearly fee can be avoided by running on the IBM AT. The yearly fee does entitle the user to SLAM II program updates. If these are installed then the advantage of paying the yearly fee is always having the latest version of SLAM II. Updates may also be purchased for the SLAM PC disks when changes are made. These updates have been \$100 to \$200. The cost per user of a mainframe version of SLAM II is less than a PC version if many users exist. In this case, however, the cost was being borne only by the department developing the Space Transportation Model. The cost of maintaining SLAM II was one of the motivations for NASA to make the comparison study.

Running jobs on the IBM AT provides predictable turn around times. The ST Ops Model runs in 6.3 minutes. It will run in the same amount of time with each execution. The execution times for all SLAM II problems run on the IBM AT were reasonably

fast. There are no indefinite waits for results because the IBM AT is run independently from the mainframe. Even if the IBM AT breaks down it is a simple matter to load the files from backup disks onto another IBM AT. The creation of the batch job SLAMII.BAT makes it simple to execute a SLAM II simulation on the IBM AT.

When a simulation has more than one output report the batch job puts these in separate files. These files are not directly readable but must be examined using the SLAM II OUTPUT processor. The processor allows specification of which part of the summary report is wanted as well as where it is to be sent. This can be somewhat inconvenient since each file must be called separately for processing, then stored in different files for later use. On the other hand, it may not be necessary to examine all summary reports. Having them all in one place, as when the mainframe is used, may also be inconvenient. That is why the batch job automatically only processes one output report. Another batch job, called REPORTS.BAT, has been created which helps to overcome the problem of repeated calls to the OUTPUT processor. The job is described and listed in Appendix C. This batch job is called with a list of all the output report files that are wanted. Each file is then put through the OUTPUT processor and the summary reports are stored together in a file called RESULTS. RESULTS may then be examined and printed.

The minimum possible execution time on the AT is greater than on the mainframe. The AT execution times experienced however,

have all been very reasonable. No run on the mainframe has executed in its minimum possible time. There have always been delays due to resource contention. The best mainframe turn around times experienced are comparable to those of the AT but, may be much larger.

The smaller storage and dimension limits presented by the AT compared to the mainframe have occasionally been a problem. This is due to a sector limitation bug in Microsoft FORTRAN. It is anticipated that once this bug is corrected, the ST Ops Model could grow to several times its current size and still not create any limit problems on the AT. In general, very large simulation models will still need a mainframe computer; but, most simulations can be handled by hard disk sized microcomputers.

Making the decision to run SLAM II simulations on an AT means that new procedures must be learned. DOS and some sort of editor will be needed. Anyone with an AT will have to learn these procedures eventually.

In summary, the advantages of using an AT outweigh the disadvantages. Also, the disadvantages of using a mainframe computer can largely be overcome on the AT without sacrificing the mainframe's strong points. Turn around times on the AT were consistently better than on the mainframe, storage space was not a problem on the AT, and convenience was maintained by using a batch job to perform the PC SLAM II execution steps.

While the results of this study are for the specific problem at NASA Langley described, the conclusions are applicable to many

circumstances and would be valid in other mainframe vs. microcomputer situations.

Acknowledgements

This work was performed at NASA Langley for W. Douglas Morris of the Space Systems Division. The Space Transportation System Operations Model was written by Mr. Morris and Mrs. Nancy White. They supplied Figures 1 and 2. NASA generously furnished the resources and support which made this study possible.

APPENDIX A.

A Batch Job to Run SLAM on the IBM AT

The size of the SLAMII program prohibits it from being put on one double sided, double density disk. If user written FORTRAN routines are necessary in the SLAM model being run then even more space is needed for the Microsoft FORTRAN and link files. Eight disks may be used to run a SLAM program with user written FORTRAN: three SLAM disks, four FORTRAN disks and at least one disk for the SLAM model and it's output files.

The use of the IBM AT allows all the necessary files to be placed on it's hard disk. This eliminates the need for disk swapping and allows a batch job to be written which will automatically perform almost all the steps necessary to get output from the PC version of SLAM.

To use the batch job that will be described a file directory called SLAM needs to be created. This directory contains all the files necessary to run SLAM programs including the batch job itself. A listing of the files is given in Figure 3 along with their functions. The simulation program files should be put in a file directory called SLAMJOBS.

The names given to the SLAM program files must follow a certain format. A simulation program may have up to three files, each with a *filename* that is the same and may be up to 8 characters long. The three files are distinguished by their extenders. One file contains the SLAM control statements and the network statements if the program has a network. This file has the extension .SLM. The second file, with extension .FOR, contains the user written FORTRAN including a new program MAIN if changes to MAIN are necessary. If there are no user written FORTRAN routines then this file is not needed. The FORTRAN routines may read data from the third file, defined with the extension .DAT. If this data file is to be used it must be opened by a FORTRAN routine in *filename* .FOR.

The batch job to run SLAM is called **SLAMII.BAT**. A listing is shown in Figure B1. **SLAMII.BAT** should be in the SLAM directory. It can be run from either the SLAM directory or the SLAMJOBS directory by creating another file also called **SLAMII.BAT** containing the statement \SLAM\SLAMII %1 and putting it in the SLAMJOBS directory. In addition, the SLAM file PRCTL.FOR should be copied in directory SLAMJOBS. This contains the Microsoft FORTRAN meta commands that will be needed if the

the SLAM file PRCTL.FOR should be copied in directory SLAMJOBS. This contains the Microsoft FORTRAN meta commands that will be needed if the simulation is run from SLAMJOBS. Put the statement

```
$INCLUDE'PRCTL.FOR'
```

beginning in column one as the first statement of the FORTRAN file.

SLAMII.BAT uses a program called KEYFAKE [Petzoid 1985] in order to store responses that will be asked for by SLAM processors. These responses will be made automatically as if they done been keyed in when called for.

APPENDIX B.

The SLAMII.BAT job

The job that executes a SLAM simulation, **SLAMII.BAT**, is executed by the user by typing

SLAMII *filename*

where *filename* is the name given to the set of up to three files that contain the SLAM job. These are located in directory SLAMJOBS. *Filename* should have no extender. **SLAMII.BAT** may be run from either of the directories SLAM or SLAMJOBS. Upon completion the current directory will be SLAMJOBS.

What **SLAMII.BAT** does can be followed by looking at Figure B1. Once **SLAMII.BAT** begins a message will appear on the screen with a reminder of the necessary format of the job's file names. After the user strikes any key the simulation begins. First, SLAMJOBS is made the current directory. The KEYFAKE program is called to set up two responses to the SLAM input processor. These are *filename* .SLM to tell the processor where the job is and *filename* .TRA which gives a file in which to put the translated model created by INPUT.EXE. After INPUT.EXE is called and executed the batch job looks in directory SLAMJOBS for a FORTRAN file under the name *filename* .FOR. If one is not found then it is assumed there are no user written FORTRAN routines and the batch job will start the EXECUTIO.EXE processor after calling KEYFAKE to set up the responses that will be asked for. These responses are *filename* .TRA for input to the execution processor and *filename* .1 for the output file. If more than one output file is called for, as would be the case with multiple SLAM summary reports, they would be put in files named *filename* .2, *filename* .3, etc., up to *filename* .12. If more output files are needed the execution will stop and the user will be asked to supply file names in which to put the results.

At this point **SLAMII.BAT** deletes *filename* .TRA which is no longer needed. The only files left created by **SLAMII.BAT** are output files and any files containing plots generated by SLAM. These output files are in the directory SLAMJOBS.

The OUTPUT.EXE processor is called using as input *filename* .1 set up by another call to KEYFAKE. A menu generated by OUTPUT.EXE will appear so that the user may select what simulation results are to be seen on the screen or transmitted elsewhere. If more than one output file has been created type \SLAM\OUTPUT for each one desired in order to see the results or use REPORTS.BAT (see Appendix C).

If a file named *filename* .FOR was found in directory SLAMJOBS after the INPUT.EXE processor finished then the FORTRAN routines are compiled. The result is an object file which is linked with the MAIN.OBJ file already in the directory SLAM. An EXE file is created and linked with the necessary library routines, SLAM32.LIB, FORTRAN.LIB and 8087.LIB. Now the execution processor is ready to be called. It will be started by calling *filename* .EXE. The remainder of the batch job behaves the same as described above for network only models including the ability to save automatically up to 12 output files and the deletion of unnecessary files.

```

ECHO OFF
REM This job automatically runs all the SLAM and FORTRAN
REM files necessary to get completed SLAM output.
cls
ECHO                WELCOME TO SLAM
ECHO -----
ECHO This program assumes that all the job files ready for
ECHO SLAM execution are on the hard disk in directory
ECHO \SLAMJOBS. The SLAM control statements and network
ECHO specification should be in file name.SLM. If there
ECHO is user written FORTRAN it should be in file
ECHO name.FOR. If the FORTRAN routines read data that
ECHO data should be in file name.DAT. In each case the
ECHO "name" should be the same. If there is a data file
ECHO read by FORTRAN routines it needs to be opened
ECHO within name.FOR before being referenced.
ECHO -----
PAUSE
CD\SLAMJOBS
\SLAM\KEYFAKE "%1.SLM" 13 "%1.TRA" 13
\SLAM\INPUT
IF NOT EXIST %1.FOR GOTO EXECU
\SLAM\FOR1 %1.FOR;
\SLAM\PAS2
\SLAM\LINK %1+\SLAM\MAIN,\SLAM\%1,,\SLAM\SLAM32+\SLAM\FORTRAN+\SLAM\8087
\SLAM\KEYFAKE "%1.TRA"13"%1.1"13"%1.2"13"%1.3"13"%1.4"13"%1.5"13"%1.6"13"%1.7"
13"%1.8"13"%1.9"13"%1.10"13"%1.11"13"%1.12"13"%1.13"13
\SLAM\%1.EXE
DEL \SLAM\%1.EXE
DEL \SLAM\%1.MAP
DEL %1.OBJ
GOTO OUTPUT
:EXECU
\SLAM\KEYFAKE "%1.TRA"13"%1.1"13"%1.2"13"%1.3"13"%1.4"13"%1.5"13"%1.6"13"%1.7"
13"%1.8"13"%1.9"13"%1.10"13"%1.11"13"%1.12"13"%1.13"13
\SLAM\EXECUTIO
:OUTPUT
DEL %1.TRA
\SLAM\KEYFAKE "%1.1" 13
\SLAM\OUTPUT

```

Figure B1. Batch Job Listing for SLAMII.BAT.

APPENDIX C.

A Batch Job to Put Summary Reports in One File

The batch job **REPORTS.BAT** has been created to automatically put output files through the SLAM OUTPUT processor and to store the resulting summary reports together in a file named **RESULTS**. The files used are created by SLAM by the execution stage of the simulation run. Each file created by this stage must be called by the OUTPUT processor in order to be put into a form which is readable. The form of the call to **REPORTS.BAT** is

```
REPORTS  filename filename ... *
```

For example, suppose as simulation as created three output files after execution, **OUT.1**, **OUT.2**, and **OUT.3**. If each one is to be put through the OUTPUT processor so that the summary reports are readable and then all three summary reports are to be put in one file for examination this can be done with the statement

```
REPORTS OUT.1 OUT.2 OUT.3 *
```

Now file **RESULTS** will contain all three summary reports.
The job **REPORTS.BAT** is listed in Figure C1.

```
ECHO OFF
REM This job puts the files listed through the SLAM
REM  OUTPUT processor and stores them together in a
REM  file called "results".
DEL RESULTS
:REPEAT
\SLAM\KEYFAKE "%1" 13 "1" 13 "2" 13 "OUT" 13 "12" 13
\SLAM\OUTPUT
TYPE OUT>>RESULTS
SHIFT
IF NOT %1 == * GOTO REPEAT
DEL OUT
```

Figure C1. Batch Job Listing for REPORTS.BAT.